

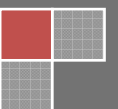
2010

SDLC

Software Development Life Cycle

Project တစ်ခုကို လက်တွေ့ အကောင်အထည်ဖော်ရာတွင် လိုအပ်သည့် အချက်များ

Pyae Phyo Maung Maung
B.C (Tech) (Hons)
MCTS , MCPD
POSTED DATE : 10 SEP 2010



SDLC (Software Development Life Cycle)

SDLC ကို မပြောပြမီ ကျွန်တော် စဉ်းစားမိသော မေးခွန်းတစ်ခု ရှိသည် ။ ကျွန်တော်တို့ Developer ဆိုသော ခေါင်းစဉ်နှင့် ဘယ် နှစ် ဘယ် အတိုင်းအတာထိ ရပ်တည်ရန် စိတ်ကူးထားပါသလဲ ။ လုပ်ငန်းတိုင်းတွင် ရာထူးတိုးသည် ဆိုသော စကားလုံးရှိလျှင် ကျွန်တော်တို့ Web လောကတွင်လည်း ထို အရာသည် ရှိကိုရှိရမည် ဖြစ်ပါသည် ။ တကယ်တမ်း စဉ်းစားကြည့်လျှင် ကျွန်တော်တို့ အထက်တွင် SA (System Analyst) ၊ Project Manager (PM) စသော အဆင့် များ ကျန်ရှိ နေပါ သေး သည် ။ ထိုမျှမက နိုင်ငံတကာတွင် Software Engineer (SE) Level ဆိုသော အဆင့်တစ်ခု အထိပင် တက်လှမ်းနိုင်ပါသည် ။

ထိုသို့ အကြောင်းအရာတစ်ခုတည်းကိုသာ အာရုံစိုက်နေသည့်အတွက် Project တစ်ခုကို လက်တွေ့တွင် အကောင်အထည်ဖော်ဆောင်ရွက်နေသည့် နည်းလမ်းများကို မေ့လျော့သွားမည် စိုးသဖြင့် ဒီ Post ကို ရေးသားဖြစ်သွားခြင်း ဖြစ်ပါသည် ။ ကျွန်တော်တို့ Developer များ အနေဖြင့် မဖြစ်မနေ စဉ်းစားကြည့်ရမည့် အချက်တစ်ခုရှိသည် ။ ကျွန်တော်တို့ အထက်တွင် အလုပ်လုပ်နေသော PM (Project Manager) ၊ SA (System Analyst) တို့ ဘာတွေလုပ်နေကြပါသလဲ ။ သူတို့ကတော့ အဓိကအားဖြင့် Domain Knowledge အသားပေးကာ လေ့လာနေကြပါလိမ့်မည် ။ တနည်းအားဖြင့် ဘယ် Project သည် ဘယ် Flow ရှိသည် ။ ဘယ် Project ကိုတော့ ဘယ်လို ဆောင်ရွက်မည် စသဖြင့် များစွာသော Project တို့နှင့် သက်ဆိုင်သည့် Domain Knowledge များကို လေ့လာနေကြပါလိမ့်မည် ။ Developer အဆင့်တွင် လေ့လာနည်းများ (Platform တစ်ခုပေါ်တွင် ထွက်သမျှ အသစ်အဆန်း နည်းပညာရပ်များ ကို လေ့လာခြင်း) ကဲ့သို့ မဟုတ်တော့ဘဲ SA / PM အဆင့်များတွင် ကွဲပြားသော Project Type များ ၏ Domain Knowledge များ ၏ ယေဘုယျ သဘောတရားများကို လေ့လာကြရမည် ဖြစ်သည် ။ ထို့ကြောင့် ကျွန်တော်တို့ လောကတွင် အထက်ရောက်လေ Programming နှင့် ပေးကွာလေ ဖြစ်ကြရမည် ဖြစ်ပါသည် ။ ကျွန်တော်တို့ တစ်ချိန်ချိန် တွင် ရောက်လာနိုင်သော Level များအတွက် ကြိုတင်ပြင်ဆင်နိုင်စေရန် ဖြစ်ပါသည် ။

တကယ်တမ်းတွင် အကြမ်းဖျင်းပြောပြရလျှင် PM (Project Manager) တစ်ယောက် အတွက် Domain Knowledge သည် အသက်ဖြစ်ပြီး SA အတွက် System တစ်ခု၏ Design ရေးဆွဲပုံ ၊ System ကြီး တစ်ခုလုံး အဆင်ပြေချောမွေ့စွာ အလုပ်လုပ်နိုင်ရန်သည် အသက်ဖြစ်သည် ။ ထို့ကြောင့် ကျွန်တော်တို့

တစ်ချိန်ချိန်တွင် ရောက်လာမည် ဖြစ်သော SA ၊ PM တို့ ၏ လုပ်ငန်းစဉ်များမှ အခြေခံ အကျဆုံးသော အရာဖြစ်သည့် SDLC ကို ကြိုတင်လေ့လာနိုင်ရန် ဖော်ပြပေးမည် ဖြစ်ပါသည်။

SDLC ဆိုသည့် အရာမှာ တကယ်တမ်းပြောရလျှင် အလွန်ကျယ်ပြန့်ပါသည် ။ Company တိုင်းတွင် ကိုယ်ပိုင်ဟန် ကိုယ်ပိုင်မူဖြင့် ကျင့်သုံးသော SDLC များစွာရှိပါသည် ။ SDLC ကို ချဉ်းကပ်ပုံ Methodologies များ ၊ SDLC Model များစွာ ရှိသည့် အနက်မှ အခြေခံအကျဆုံး အချက်များကိုသာ ရွေးချယ်တင်ပြထားခြင်း ဖြစ်ပါသည် ။ ကျွန်တော် အကုန်နီးပါးရေးပြလျှင် ရှုပ်ထွေးပြီး ပြောချင်သော အချက်များကို မထိမိမှာ စိုးရိမ်သောကြောင့် ဖြစ်ပါသည် ။

Project တစ်ခုကို လက်တွေ့ မိမိကိုယ်တိုင် အကောင်အထည်ဖော် ရမည့်သူများ ၊ Programming တစ်ခုတည်းကိုသာ အာရုံစိုက်နေသူများ အတွက် လက်တွေ့တွင် အရေးပါဆုံး ဖြစ်သော SDLC ၏ အခြေခံအကျဆုံး အချက်အလက်များကို မေ့လျော့နေခြင်းမျိုး မဖြစ်ပေါ်စေရန် ရည်သန်ပြီး ရေးသားလိုက်ခြင်း ဖြစ်ပါသည် ။ ကျွန်တော် ပြောပြမည့် SDLC ၏ အချက်အလက်များကို အခြေခံပြီး **SDLC အကြောင်းကို အသေးစိတ် ဆက်လက်လေ့လာရန်လည်း မျှော်လင့်မိသည့်အတွက် ဖြစ်ပါသည် ။**

Company Project နှင့် မိမိကိုယ်တိုင် အစအဆုံးတာဝန်ယူရသော Project

ကျွန်တော်တို့ Programming ကို လေ့လာကြသည် ။ Programming Languages , Programming Techniques , Programming Concept များနဲ့သာ မေ့လျော်ရင်း အချိန်တော်တော်များများ ကုန်ဆုံးခဲ့ကြသည်။ Programmer အချင်းချင်းတွေလျှင် Coding အကြောင်းဆွေးနွေးကြသည် ။ Coding ကို သာ အာရုံစိုက်နေရင်း တကယ်တမ်း လက်တွေ့တွင် Project တစ်ခု အကောင်အထည်ဖော်ရာတွင် မရှိမဖြစ် ၊ မသိမဖြစ် အရေးကြီးသော အစိတ်အပိုင်းများကိုပင် မေ့လျော့နေတတ်ကြသည် ။ ကျွန်တော်တို့ စူးစူးစိုက်စိုက် လေ့လာနေသော Coding ဆိုသည့် ဘယ်နေရာမှာ ရှိနေပါသလဲ ။ တကယ်တမ်း Project တစ်ခုကို လက်တွေ့ အကောင်အထည်ဖော်ရာတွင် Coding ထက် အရေးကြီးသော အစိတ်အပိုင်း များစွာ ရှိပါသည် ။

Company Project

ကျွန်တော် Company မှာတုန်းက အလုပ်လုပ်ရသည်မှာ ပျော်စရာ ။ Project Manager က ဆုံးဖြတ် SA က ဒီဇိုင်း ဆွဲ ၊ ထွက်လာတဲ့ ဒီဇိုင်းကို အခြေခံပြီး ကျွန်တော်တို့ Programmer များက Coding ရေး ဒီလိုနဲ့ နေသားတကျ ရှိခဲ့သည် ။ Customer ဆီမှ System Error တက်သည်ဟုဆိုလျှင် SA က အရင်စစ်မည် ။ ထို့နောက် ဘယ်နေရာ ဘာတွေ ဖြစ်နေသည်ဟု SA က Developer ကို လာပြောမည် ။ ကျွန်တော်တို့

Programmer များက ပြင်ပေးရုံ တာဝန် သက်သက်သာ ဖြစ်သည်။ SA ဘာလုပ်နေသည်ကို ကျွန်တော်တို့ မမြင်။ တကယ်လမ်း လေ့လာစရာမလိုအပ်ဟု ခံယူရင်း ကိုယ့်နည်းပညာရပ်ကိုဘက်ကိုသာ အာရုံစိုက် နေကြတော့သည်။ Coding သာလျှင် အဓိ၊ Coding သာလျှင် အဖ ဖြစ်ခဲ့ရသည်။ Coding ရေးနေရလျှင် အရာရာပြည့်စုံနေတော့သည်။

“ လူတို့သည် အခွင့်အရေးကိုမမြင်။ အဘယ်ကြောင့်ဆိုသော် အခွင့်အရေး ဟူသည် ခက်ခဲသော အသွင်တစ်ခု အဖြစ် ပင်ရောက်လာတတ်သည် ” ဆိုသော ဆောင်ပုဒ်လေးကို လက်ကိုင်ထားသော ကျွန်တော် တကယ်တမ်း Project လေးတစ်ခုကို လက်ခံရသောအခါ လက်စွမ်းက ပြချင်လာသည်။ အတွေ့အကြုံသည် တကယ်တမ်း လက်တွေ့တွင် အလွန်စကားပြောပါသည်။ ကျွန်တော် စလုပ်ရသော Project တစ်ခုကို ကျွန်တော် နှင့် သူငယ်ချင်းတစ်ယောက်ပေါင်းကာ လက်ခံမိသည်။ Coding ကိုသာ ကျွမ်းကျင်သော ကျွန်တော်တို့အတွက် တကယ်တမ်း လက်တွေ့တွင် Project တစ်ခု ဖန်တီးသောအခါ SDLC က ပညာ စပြပါတော့သည်။ တကယ်တမ်း Project တစ်ခုကို Company တစ်ခု ၊ အသင်းအဖွဲ့ တစ်ခု အနေနဲ့ ဘယ်လိုများ အကောင်အထည်ဖော်ကြပါသလဲ ဆိုသည်ကို မစဉ်းစားခဲ့သမျှ အခုတော့ အခက်အခဲများစွာ တွေ့ရတော့သည်။

ကျွန်တော် မစဉ်းစားမိသည့် အချက်မှာ ကျွန်တော်တို့ Project တစ်ခုကို လက်ခံ ရရှိသော အချိန်တွင် မည်သည့်နေရာမှ စတင်ဆောင်ရွက်ရန် စိတ်ကူးထားမိပါသနည်း ဆိုသော အချက်ပင် ဖြစ်သည်။ Company များတွင် အလုပ်လုပ်နေသော သူများအတွက် ယခုဖော်ပြသော အချက်များမှာ Project Manager ၊ SA (System Analyst) ၏ အလုပ်များသာ ဖြစ်ပါသည် ဟု ဆင်ခြေကန် စရာရှိသော်လည်း တကယ်တမ်း လက်တွေ့တွင် Project တစ်ခုကို မိမိကိုယ်တိုင် ဆုံးခန်းတိုင် အကောင်အထည် ဖော်ရမည့် အခြေအနေမျိုးနှင့် ကြုံတွေ့တတ်ရတတ်သည်မှာ Programmer (Developer) တို့ ၏ အစဉ်အလာပင် ဖြစ်နေတတ်သည်။ ငွေကြေး အနည်းအများ ပေါ် မူတည်ပြီး လုပ်ရသည့် အပိုင်းများ အနည်းငယ် ကွာခြားသည်သာ ရှိသော်လည်း တကယ်တမ်းတော့ Project မှာ Project သာ ဖြစ်သည်။

ထို့ကြောင့် ကျွန်တော်ပေးလိုသော Information တစ်ခုမှာ Project တစ်ခု Develop လုပ်နိုင်ရန် အတွက် မရှိမဖြစ် အရေးပါ အရာရောက်သော SDLC ကို မေ့မထားကြရန် သတိပေးလိုခြင်း ဖြစ်ပါသည်။

အရင်က SDLC ကို (Software Development Life Cycle) ဟုခေါ်ဝေါ်သုံးစွဲ ခဲ့ရာမှ ယခုအခါ SDP (Software Development Process) ဟု ပြောင်းလဲ ခေါ်ဝေါ်သုံးစွဲ လာကြသည်။

SDLC တွင် ပါဝင်သော အကြောင်းအရာများစွာရှိသော်လည်း ထိုအချက်များထဲမှ အခြေခံ အကျဆုံး အရေးပါဆုံးအချက်များသာ တင်ပြမည် ဖြစ်ပါသည်။ ။ ထိုအချက်များမှာ

- 1 . Requirement Definition
- 2 . Requirement Analysis
- 3 . Requirement Specification
- 4 . Design
- 5 . Implementation (Coding)
- 6 . Testing
- 7 . Delivery
- 8 . Maintenance

စသည်တို့ တင်ပြမည် ဖြစ်ပါသည်။ ။

1 . Requirement Definition

Project နှင့် ပတ်သက်ပြီး Customer လိုအပ်သော အကြောင်းအရာ အချက်အလက်များကို စတင်စုဆောင်းသည့် အဆင့်ဖြစ်ပါသည်။ ။

2 . Requirement Analysis

Customer ထံမှ ရလာသော အကြောင်းအချက်အလက်များကို ကျွန်တော်တို့ ပြန်လည် ဆန်းစစ်ရန် လိုအပ်သည့် အဆင့် ဖြစ်သည်။ ။

3 . Requirement Specification

Customer ဆီမှ ရရှိလာသော အကြောင်းအရာ အချက်အလက်များကို လက်ခံစဉ်းစားပြီး ဖြစ်သင့်ဖြစ်ထိုက်သော အချက်အလက်များကို Customer နှင့် ညှိနှိုင်းဆွေးနွေးရမည့် အဆင့်ဖြစ်သည်။ ။

4 . Design

ထို့နောက် Design ပိုင်းကို စဉ်းစားရမည် ဖြစ်သည်။ Design ပိုင်းတွင် အနည်းဆုံး အပိုင်း(၃) ပိုင်း ပါဝင်အောင် စဉ်းစားရမည် ဖြစ်ပါသည်။

- External Design (Logical Design)

Logical Design ခေါ် ပုံကြမ်းရေးဆွဲပါ။ ။ ရလာသော ပုံကြမ်းကို Customer အား ပြသပြီး အတည်ပြုချက် ရယူပါ။ ။ ပုံကြမ်းကို Customer နှစ်သက်ရန် လိုအပ်ပါသည်။ ။ ဒီ အဆင့်သည် Documentation ပိုင်းမျှသာ ဖြစ်ပါသည်။ ။

- Prototype

ပုံအကြမ်းထည် အဆင့်မှ ပါဝင်သော အချက်အလက်များကိုစုစည်းပြီး Prototype တစ်ခုအသွင် Software အသုံးပြုပြီး ရုပ်လုံးပေါ်အောင်ရေးဆွဲပါ။ ။ ရလာသော Prototype ကို Customer အား ပြသပြီး အတည်ပြုချက် ရယူပါ။ ။

- Internal Design (Physical Design)

Prototype အသွင်မှ အသေးစိတ် Design ပေါ်အောင် ဆက်လက်ရေးဆွဲပါ။ ။ Customer သဘောတူညီမှုကို အလေးအနက်ထား ရယူပါ။ ။

5 . Implementation (Coding)

ရေးဆွဲပြီးသော Design တစ်ခုပေါ်မူတည်ပြီး Coding ရေးသားခြင်း အပိုင်းကို လုပ်ဆောင်ပါ။ ။

6 . Testing

အရေးကြီးသော အစိတ်အပိုင်းတစ်ခု အဖြစ် Testing ကို အလေးထားလုပ်ဆောင်ပါ။ ။ မိမိ Software ကို Customer နေရာမှ နေပြီး အထပ်ထပ် အခါခါ စမ်းသပ်ပါ။ ။ ဗဟုသုတ အနေနှင့် ပြောရလျှင် Company အများစုတွင်တော့ Testing ပိုင်းတွင် White Box Testing နှင့် Black Box Testing တို့ကို ပြုလုပ်လေ့ရှိ ကြပါသည်။ ။

7 . Delivery

Customer အား စိတ်ကြိုက် အသုံးပြုခွင့်ပေးပါ။ ။ ထိုသို့ပေးသုံးသောကာလကို စမ်းသပ်ကာလ ဟု သတ်မှတ်လေ့ရှိပြီး ထိုကာလအတောအတွင်း ဖြစ်ပေါ်လာသော အမျိုးမျိုးသော Error များအတွက် ဝန်ဆောင်ခ မယူပဲ Company များ ဘက်မှ ပြုပြင်ပေးလေ့ ရှိသည်။ ။ Software တစ်ခု (သို့) Web Site တစ်ခုအတွက် စမ်းသပ်သုံးစွဲခွင့် ကာလ ကို Company များက လ အနည်းငယ် သတ်မှတ်ပေးလေ့ရှိသည်။ ။

8 . Maintenance

Maintenance ဆိုသည်မှာ Company ၏ သတ်မှတ်ကာလ ကျော်လွန်လာပြီးနောက် Customer များထံမှ တက်လာသော Demand (ထပ်မံလိုအပ်ချက်များ) များ ၊ Project နှင့် ပတ်သက်သော Error များ အတွက် Maintenance Fee (ပြုပြင်ထိန်းသိမ်းခ) ကို Company သို့ ပေးဆောင်ရန် အတွက် SDLC တွင် ထည့်သွင်းထားရခြင်းပင် ဖြစ်ပါသည်။ ။

မိမိကိုယ်တိုင် အစအဆုံး တာဝန်ယူရသော Project

ကျွန်တော် Project တစ်ခုကို စရေးတော့ SDLC ကို သေချာ သဘောမပေါက်သဖြင့် အခက်အခဲများစွာ ကြုံတွေ့ခဲ့ရသည်ကို ပြန်လည်မျှဝေလိုခြင်း ဖြစ်ပါသည်။ ။ လက်တွေ့တွင် Project တစ်ခု ကို မည်မျှတန်ပါသည် ဆိုသော စကားသည် Project များစွာ Develop လုပ်ဖူးသော သူများအတွက်ပင် အခက်အခဲများစွာရှိပါသည်။ ။ တကယ်တမ်း လက်တွေ့တွင် Project တစ်ခု ကို အကောင်အထည်ဖော်ရာတွင် SDLC ကို မေ့လျော့ခြင်းသည် အလွန်ပင် စိတ်ပင်ပန်းခြင်း ၊ မျက်နှာပျက်ရခြင်းများကို ကြုံတွေ့နိုင်သည့်အတွက် စေတနာ ရှေ့ထားကာ ဖော်ပြ ပေးလိုက်ခြင်း ဖြစ်ပါသည်။ ။ ယခု ကိုယ်လုပ်ပေးနေသော Project သေးသေးလေးအတွက် ပင် Customer အား SDLC ၏ နောက်ဆုံးအချက် (Maintenance) အကြောင်းကို မရှင်းပြမိသည့်အတွက် အခက်အခဲ များစွာ ကြုံတွေ့ နေရသူများ ၊ ယခုမှ Project တစ်ခုကို စတင်လက်ခံထားသူများ ၊ SDLC အကြောင်းကို လုံးဝ မေ့လျော့နေသူများအတွက်ပါ ဒီ Post ကို ဖတ်ပြီး နောက်တွင် Project တစ်ခုကို Develop လုပ်ရာတွင် တွေ့ကြုံ ရမည့် အခက်အခဲများအား လွယ်ကူစွာ ကျော်လွှားနိုင်ပါစေရန် ဖော်ပြလိုက်ပါသည်။ ။

ကျယ်ပြန့်လှသော SDLC ၏ အကြောင်းများကို ဆက်လက် လေ့လာနိုင်စေရန်လည်း မျှော်လင့် မိပါသည်။

ပြည်ဖိုးမောင်မောင်

ASP.Net Training Guide

B.C (Tech) (Hons :)

MCTS , MCPD .

Website :

Email :